Library function usage

Example 1. (Read and write texts to files)

```
1  /* read, write and upper-case texts of file, Kit Tyabandha, 19 Dec 06 */
2  #include<ctype.h>
3  #include<stdio.h>
4  #include<stdlib.h>
5  void
6  filerror(char s[]){
7    printf("\n Error : Can not open file `%s'\n\n", s);
8  }
9  void
10 puttext(char s[]){
11   FILE *fpt;
12   char c;
13   fpt =fopen(s, "w");
14   printf("\n Enter the contents for `%s' (backslash to end)\n\n", s);
15   do
16     putc(c=getchar(), fpt);
17   while (c!='\\');
18   fclose(fpt);
19 }
20 void
21 showtext(char s[]){
22   FILE *fpt;
23   char c;
24   if((fpt=fopen(s, "r"))==0)
25     filerror(s);
26   else{
27     printf("\n\t Contents of file `%s'\n\n", s);
28     do
29       putchar(c=getc(fpt));
30     while(c!='\\');
31   }
32   fclose(fpt);
33 }
34 void
35 uppercasetext(char s[], char S[]){
36   FILE *fpt, *fpu;
37   char c;
38   short i =0;
39   if((fpt=fopen(s, "r"))==0){
40     filerror(s); i++;}
41   if((fpu=fopen(S, "w"))==0){
42     filerror(S); i++;}
43   if(~i)
44     do
45       putc(toupper(c=getc(fpt)),fpu);
46     while(c!='\\');
47   fclose(fpt);
48   fclose(fpu);
49 }
50 int
51 main(){
52   puttext("data.txt");
53   showtext("data.txt");
54   uppercasetext("data.txt","DATA.txt");
55   showtext("DATA.txt");
56   exit(0);
57 }
```

Example 1 lists a programme that exchanges texts between standard input, standard output and a file, and transforms the same and puts in another file. The output of the programme is listed in Output 1.

Output 1 *Compilation and run of Example 1*

```
kit@nebula:~/prog/c$ make tst
gcc -c -g -Wall tst.c
gcc -g tst.o -o tst -lcurses -ldl -lm
kit@nebula:~/prog/c$ tst
 Enter the contents for 'data.txt' (backslash to end)
The road goes ever on and on
Down from the door where it began.
                J.R.R. Tolkien
                ('Lord of the Rings', 1954) \
         Contents of file 'data.txt'
The road goes ever on and on
Down from the door where it began.
                J.R.R. Tolkien
                ('Lord of the Rings', 1954) \
         Contents of file 'DATA.txt'
THE ROAD GOES EVER ON AND ON
DOWN FROM THE DOOR WHERE IT BEGAN.
                J.R.R. TOLKIEN
                ('LORD OF THE RINGS', 1954) \kit@nebula:~/prog/c$
```

Example 2 is a programme which shows how to manipulate integral arithmetic in various bases, namely decimal, hexadecimal and octal. Output 2 is its output.

Example 2. (Arithmetic of decimal, octal and hexadecimal integers)

```
#include<stdio.h>
int main(){
  short a, b, c;
  a =12;
  b =47;
  c =a+b;
  printf("\n\t Decimal\n\
    a = %d, b = %d, c = a+b = %d\n\n", a, b, c);
  a =012;
  b =047;
  c =a+b;
  printf("\n\t Decimal equivalent of octal\n\
    a = %d, b = %d, c = a+b = %d\n\
   (a = %o, b = %o, c = a+b = %o in octal)\n\n", a, b, c, a, b, c);
  a =0x12;
  b =0x47;
  c =a+b;
  printf("\n\t Decimal equivalent of hexadecimal\n\
    a = %d, b = %d, c = a+b = %d\n\
   (a = %x, b = %x, c = a+b = %x in hexadecimal)\n\n", a, b, c, a, b, c);
  return 0;
}
```

¶

Output 2 *Output of Example 2*

```
     Decimal
 a = 12, b = 47, c = a+b = 59
     Decimal equivalent of octal
 a = 10, b = 39, c = a+b = 49
(a = 12, b = 47, c = a+b = 61 in octal)
     Decimal equivalent of hexadecimal
 a = 18, b = 71, c = a+b = 89
(a = 12, b = 47, c = a+b = 59 in hexadecimal)
```

□

Example 3 shows a programme demonstrating the use of the power function `pow()`, which takes two inputs. The output is given in Output 3.

Example 3. (Usage of the power function)

```
1 /* power function, a library function, Kit Tyabandha, 20 Nov 06*/
2 #include<math.h>
3 #include<stdio.h>
4 int main(){
5   int d1, d2, p;
6   d1 =4;
7   d2 =3;
8   p =pow(d1,d2);
9   printf("\n %d to the power of %d is %d\n\n", d1, d2, p);
10  return(0);
11 }
```

¶

Output 3 *Output to Example 3*

```
4 to the power of 3 is 64
```

□

Example 4 is a programme to study some of the mathematical functions. The output is shown as Output 4. Notice how mathematical constants like Pi (π) which is infinite in size are kept on the computer as floating point constants. Also $\sin(60^\circ) = \cos(30^\circ) = \sqrt{(3)}/2$ and some hyperbolic functions, namely sinh and cosh.

Example 4. (Study of some mathematical functions)

```
1  /* mathematical functions, Kit Tyabandha, 19 Dec 06 */
2  #include<math.h>
3  #include<stdio.h>
4  int main(){
5    printf("\n Pi is %f\n", M_PI);
6    printf("\n Pi is %.9f\n", M_PI);
7    printf("\n Pi is %.60f\n", M_PI);
8    printf("\n sine(pi) = %f\n", sin(M_PI));
9    printf("\n sine(30 degree) = %f\n", sin(30*M_PI/180));
10   printf("\n cosine(30 degree) = %f\n", cos(30*M_PI/180));
11   printf("\n sine(60 degree) = %f\n", sin(60*M_PI/180));
12   printf("\n sqrt(3)/2 = %f\n", sqrt(3)/2);
13   printf("\n sinh(pi/3) = %f\n", sinh(M_PI/3));
14   printf("\n cosh(pi/3) = %f\n", cosh(M_PI/3));
15   return(0);
16 }
```

¶

Output 4 *Output of Example 4*

```
Pi is 3.141593
Pi is 3.141592654
Pi is 3.141592653589793115997963468544185161590576171875000000000000
sine(pi) = 0.000000
sine(30 degree) = 0.500000
cosine(30 degree) = 0.866025
sine(60 degree) = 0.866025
sqrt(3)/2 = 0.866025
sinh(pi/3) = 1.249367
cosh(pi/3) = 1.600287
```

□

Arrays can be multi-dimensional. An example of a three-dimensional array is demonstrated with the use of a programme in Example 5, which gives an output shown in Output 5.

Example 5. (Three-dimensional array)

```
1  /* 3-d array's addresses, Kit Tyabandha, 28 Nov 2006 */
2  #include<stdio.h>
3  int main(){
4    int i, j, k, a[2][3][4];
5    for(i=0; i<2; i++){
6      for(j=0; j<3; j++){
7        for(k=0; k<4; k++){
8          printf("&a[%d][%d][%d] = %x\n", i, j, k, &a[i][j][k]);
9        }
10     }
11   }
12   return 0;
13 }
```

¶

Output 5 *Output to Example 5*

```
&a[0][0][0] = bffffa80
&a[0][0][1] = bffffa84
&a[0][0][2] = bffffa88
&a[0][0][3] = bffffa8c
&a[0][1][0] = bffffa90
&a[0][1][1] = bffffa94
&a[0][1][2] = bffffa98
&a[0][1][3] = bffffa9c
&a[0][2][0] = bffffaa0
&a[0][2][1] = bffffaa4
&a[0][2][2] = bffffaa8
&a[0][2][3] = bffffaac
&a[1][0][0] = bffffab0
&a[1][0][1] = bffffab4
&a[1][0][2] = bffffab8
&a[1][0][3] = bffffabc
&a[1][1][0] = bffffac0
&a[1][1][1] = bffffac4
&a[1][1][2] = bffffac8
&a[1][1][3] = bffffacc
&a[1][2][0] = bffffad0
&a[1][2][1] = bffffad4
&a[1][2][2] = bffffad8
&a[1][2][3] = bffffadc
```

□